RECEIVED
CENTRAL FAX CENTER
JUL 0 5 2006

Remarks

Claims 1-6 were rejected under 35 USC 102(b) as being allegedly unpatentable in view of US patent 6,192,455 ("Boggins"). Applicant responds as follows.

It may be possible to interpret Boggins in two ways. Neither interpretation anticipates Applicant's independent claims. On the contrary, according to either interpretation, Boggins uses different structure to solve a different problem.

Under the first interpretation, Boggins' address range 127 corresponds to the conventional PCI memory address range below 4GB that Applicant refers to at the lower left side of Applicant's Fig. 2. Given any user address falling within this PCI memory address range, the CPU response will be to generate a physical address that has not been translated. In other words, the CPU's TLB will generate a "straight-through" mapping for such an address. When Boggins uses the word "reserved" in this context, he simply means that these physical addresses cannot be remapped using conventional CPU page table/TLB mechanisms. In other words, the operating system cannot allocate any physical pages in this range in response to user memory allocation requests. That is why they must be "reclaimed" if they are to be allocated for use by user programs.

According to the above interpretation, Boggins does not anticipate Applicant's independent claims because Boggins does not attempt to reclaim this region of physical memory, let alone reclaim it in a way that leaves part of it unreclaimed.

Under the second interpretation, Boggins' address range 127 corresponds to a conventional "remap window." In order to do PCI memory reclaiming according to the prior art, a "remap window" of user addresses is established above the top of physical memory (see upper left side of Applicant's Fig. 2 and Applicant's description at page 3, lines 1-15). The result of establishing this remap window is that the CPU will generate "physical" addresses above 4GB in response to user addresses pointing into the remap window. These "physical" addresses above 4GB are translated *again* inside the chipset so that they are aliased to the physical PCI memory range, thus accomplishing the desired reclamation (see Applicant's Fig. 2 at 202).

But if Boggins' address range 127 is taken to mean the conventional remap window just described, then the apparatus of Boggins still leaves the system exposed to the AGP aperture bug

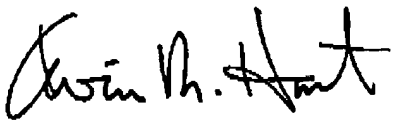
described by Applicant at page 3, lines 16-22: Unlike the remap windows of the prior art, and unlike the second interpretation of Boggins, Applicant teaches creating a disabled portion of the remap window (see range 2 at the upper left side of Applicant's Fig. 2). Applicant does so by reporting to the operating system at boot time that a portion of the remap window is reserved. The consequence is that *no* physical memory addresses above 4GB will be generated by the CPU that would be aliased by the chipset to the physical AGP aperture. In Boggins, no part of the address range 127 is disabled. Consequently, in Boggins, physical addresses above 4GB *will* be generated by the CPU that would be aliased by the chipset to the physical AGP aperture.

Under the second interpretation, therefore, Boggins does not anticipate Applicant's independent claims because Boggins does not disable any portion of the remap window. Instead, Boggins leaves all of the remap window enabled and simply catches GTLB misses inside the chipset to make certain that no user program can access the SMRAM area. Doing so does not address the AGP aperture bug that Applicant's invention solves.

Conclusion

For at least the above reasons, Applicant asserts that all independent claims are allowable over the prior art of record. Applicant's dependent claims are allowable for at least the reason that they depend from an allowable independent claim.

Respectfully submitted,



Date: 6/28/2006

Kevin M. Hart
Reg. No. 36,823